

Calculadora de Notas

December 17, 2023



1 Calculadora de Notas - Sistema de Avaliação UVV

Este projeto, intitulado Calculadora de Notas, é uma implementação em Python que permite o registro e cálculo das notas de alunos de um curso EAD da Universidade Vila Velha (UVV). Desenvolvido por Pablo Alves Correia, estudante de Ciência de Dados (Matrícula: 201312840), este sistema foi criado como parte de uma atividade acadêmica da disciplina Introdução a Linguagem de Programação.

Descrição do Sistema O código proporciona uma funcionalidade de entrada e validação de notas para cem alunos, registrando seus desempenhos em três Atividades Online Pontuadas (AOP1, AOP2, AOP3) e nas provas regulares. Além disso, oferece a opção de inserção de notas para a Prova de Recuperação, caso o aluno não atinja a média mínima para aprovação.

Funcionalidades Principais Utilização de uma função para gerenciar a entrada de notas, garantindo que valores dentro dos limites sejam inseridos. Cálculo da nota total de cada aluno e determinação do seu status: Aprovado, Prova de Recuperação ou Reprovado, de acordo com critérios predefinidos. Exibição da porcentagem de alunos aprovados e reprovados na turma. Este sistema é uma ferramenta útil para analisar o desempenho dos alunos e fornecer informações sobre o percentual de aprovação na turma.

```
[ ]: # DIÁRIO LANÇAMENTO DE NOTAS ALUNO UVV - ON // PABLO ALVES CORREIA 202312840 -
    CIÊNCIA DE DADOS

#Para otimizar o código criar uma função que lide com as entradas de notas e
    faça as validações.
def obter_nota(min_valor, max_valor, mensagem):
    while True:
        try:
            nota = float(input(mensagem))
            if min_valor <= nota <= max_valor:
                return nota
            else:
                print(f"Por favor, insira um valor entre {min_valor} e
    {max_valor}.")
        except ValueError:
            print("Por favor, insira um valor numérico.")

print('-----*-----*-----*----- DIÁRIO DE NOTAS -----*-----*-----*-----')

#Inicializando Contadores de Aprovados/Reprovados
aprovados = 0
reprovados = 0

#Loop para garantir a inserção e o range de dados
for i in range(100):
    aluno = input(f'INFORME o NOME do aluno {i+1}: ')

    notas = [
        obter_nota(0, 1, f'Nota [0, 1] na AOP01 do aluno {i+1}: '),
        obter_nota(0, 2, f'Nota [0, 2] na AOP02 do aluno {i+1}: '),
        obter_nota(0, 1, f'Nota [0, 1] na AOP03 do aluno {i+1}: '),
        obter_nota(0, 6, f'Nota [0, 6] na PROVA REGULAR do aluno {i+1}: ')
    ]

# Cálculo da Soma das notas e verificação do status
    nota_total = sum(notas)
    print(f"Nota Total até o momento: {nota_total:.2f}")

    if nota_total >= 7.0:
        status = "Aprovado"
        aprovados += 1
    else:
        prova_recuperacao = obter_nota(0, 10, f'O Aluno {aluno} NÃO atingiu a
    média para APROVAÇÃO.'

        '\nInforme a Nota da prova de
    RECUPERAÇÃO:\n')
        nota_final_recuperacao = (nota_total + prova_recuperacao) / 2
```

```

        if nota_final_recuperacao >= 5.0:
            nota_total = nota_final_recuperacao
            status = "Aprovado"
            aprovados += 1
        else:
            status = "Reprovado"
            reprovados += 1

#Exibição dos resultados calculados
print(f"Aluno {i+1}:", aluno)
print(f"NOTA FINAL: {nota_total:.2f}")
print(f"Status do Aluno: {status}")

# Porcentagem de aprovados x reprovados
porcentagem_aprovados = (aprovados / 100) * 100
porcentagem_reprovados = (reprovados / 100) * 100

# Percentual de alunos aprovados e reprovados na turma
print(f"Porcentagem de alunos APROVADOS na turma é de: {porcentagem_aprovados:.2f}%")
print(f"Porcentagem de alunos REPROVADOS na turma é de: {porcentagem_reprovados:.2f}%")

```

1.0.1 Conclusão: Reflexões sobre o Desenvolvimento do Sistema de Avaliação UVV

A criação da Calculadora de Notas, um sistema de avaliação para os alunos do curso EAD da UVV, foi uma experiência enriquecedora do ponto de vista do desenvolvimento em Python e da lógica por trás de sistemas de pontuação acadêmica.

Ao longo deste projeto, aprendi a importância da estruturação do código e a modularização por meio de funções para garantir um fluxo eficiente de entrada de dados, validação e cálculos. A criação da função obter_nota para gerenciar as entradas e as validações demonstrou como é fundamental garantir a integridade dos dados inseridos pelo usuário.

Além disso, o processo de determinar o status do aluno com base nas notas obtidas em diferentes avaliações e nas condições de aprovação ou recuperação destacou a necessidade de compreender profundamente os requisitos do sistema. Isso foi fundamental para oferecer resultados precisos e funcionais.

Um ponto de destaque foi a importância da lógica de programação na resolução de problemas. A implementação das condições para determinar o status do aluno exigiu uma análise cuidadosa dos critérios estabelecidos e a aplicação de lógica condicional adequada.

Por fim, a experiência de desenvolver este sistema proporcionou não apenas conhecimento técnico em Python, mas também uma compreensão mais profunda sobre a importância da clareza nos requisitos e da organização do código para alcançar um programa funcional e eficiente.

Este projeto serviu como um ponto de partida valioso para explorar a implementação de sistemas similares e reforçou a importância do planejamento e da estruturação para o sucesso de qualquer

projeto de desenvolvimento.

1.1 FEEDBACK AVALIATIVO DA ATIVIDADE

Após receber o feedback do professor pela execução desta atividade, desenvolvi este código para exibir as avaliações em formato de imagem. Este trabalho resultou na conquista da nota máxima e contribuiu significativamente para minha aprovação na disciplina.

O código a seguir cria uma janela que apresenta duas imagens lado a lado, refletindo os feedbacks recebidos no Ambiente Virtual de Aprendizagem (AVA), fornecendo uma visão visual das avaliações realizadas durante a execução deste projeto.

```
[2]: import matplotlib.pyplot as plt
import matplotlib.image as mpimg

# Caminhos para as imagens
caminho_imagem_1 = r'C:\PYTHON\PORTIFÓLIO\UVV_CALCULADORA_NOTAS\Feedback01.jpg'
caminho_imagem_2 = r'C:\PYTHON\PORTIFÓLIO\UVV_CALCULADORA_NOTAS\Feedback02.jpg'

# Carregar as imagens
img1 = mpimg.imread(caminho_imagem_1)
img2 = mpimg.imread(caminho_imagem_2)

# Configurar a exibição em subplots
plt.figure(figsize=(10, 5))

plt.subplot(1, 2, 1)
plt.imshow(img1)
plt.title('Feedback 01')
plt.axis('off')

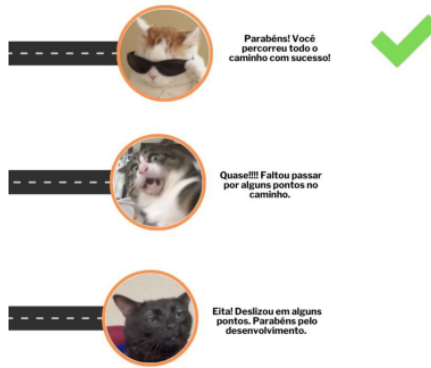
plt.subplot(1, 2, 2)
plt.imshow(img2)
plt.title('Feedback 02')
plt.axis('off')

plt.tight_layout() # Garante o espaçamento adequado entre as imagens
plt.show()
```

Feedback 01



Feedback 02



Pablo Alves Correia. Vila Velha 17 Dezembro 2023